

In the Claims

This listing of claims will replace all prior versions and listings of claims in the application:

1 1. (Previously Presented) A pipelined data processor
2 operating in a plurality of pipeline phases including at least an
3 instruction decode pipeline phase and an execution pipeline phase
4 capable of predicated instruction execution dependent upon the
5 state of an instruction designated predicate register comprising:

6 a data register file including a plurality of read/write,
7 general purpose data registers;

8 an instruction decode unit operative during said instruction
9 decode pipeline phase receiving fetched instructions and
10 determining the identity of at least one source operand data
11 register, a destination operand data register and one of a
12 plurality of functional units for execution of each instruction,
13 said instruction decode unit further identifying a predicate
14 register responsive to receipt of a predicated instruction;

15 a scoreboard bit corresponding to each data register capable
16 of serving as a predicate register, each scoreboard bit connected
17 to said instruction decode unit to be set to a first digital state
18 upon determining said corresponding data register is a destination
19 for an instruction and connected to said plurality of functional
20 units to be reset to a second digital state opposite to said first
21 digital state upon functional unit write of a result to said
22 corresponding data register;

23 said plurality of functional units operative during an
24 execution pipeline phase connected to said instruction decode unit
25 for performing a data processing operation on at least one source
26 operand recalled from at least one corresponding instruction
27 designated source data register and producing a result, said
28 functional unit

responsive to an instruction not a predicate instruction to write said result to an instruction designated destination data register, and

responsive to a predicate instruction to write said result to an instruction designated destination data register if said corresponding predicate data register has a first state during said execution pipeline phase regardless of said state of said corresponding scoreboard bit and to nullify said instruction and not write said result if said predicate register has a second state opposite to said first state during said execution pipeline phase regardless of said state of said corresponding scoreboard bit; and

each functional unit is further operative responsive to a predicate instruction during said instruction decode pipeline phase to nullify said predicate instruction of a following execution phase by operating at a reduced power state relative to normal instruction operation if said predicate register has said second state during said instruction decode pipeline phase and said corresponding scoreboard bit has said second state during said instruction decode pipeline phase.

2. (Original) The pipelined data processor of claim 1, wherein:

said functional unit is further operative to reset said scoreboard bit to said second digital state upon nullification of said instruction designating a corresponding data register as a destination operand data register.

3. (Canceled)

4. (Previously Presented) A method of operating a pipelined data processor operating in a plurality of pipeline phases

3 including at least an instruction decode pipeline phase and an
4 execution pipeline phase capable of predicated instruction
5 execution dependent upon the state of an instruction designated
6 predicate register comprising the steps of:
7 setting a scoreboard bit to a first digital state upon
8 determining a corresponding data register is a destination for an
9 instruction;
10 resetting a scoreboard bit to a second digital state opposite
11 to said first digital state upon a write of a result to said
12 corresponding data register;
13 performing a data processing operation via a corresponding
14 functional unit on at least one source operand recalled from at
15 least one corresponding instruction designated source data
16 register, producing a result and writing said result to an
17 instruction designated destination data register in response to an
18 instruction not a predicate instruction;
19 performing a data processing operation via a corresponding
20 functional unit on at least one source operand recalled from at
21 least one corresponding instruction designated source data register
22 and producing a result in response to a predicate instruction
23 designating a corresponding predicate data register and writing
24 said result to an instruction designated destination data register
25 regardless of said state of said corresponding scoreboard bit if
26 said corresponding predicate data register has a first state during
27 said execution pipeline phase;
28 nullifying a data processing operation of a predicate
29 instruction by not writing said result to the instruction
30 designated destination data register via said corresponding
31 functional unit regardless of said state of said corresponding
32 scoreboard bit if said corresponding predicate register has a
33 second state opposite to said first state during said execution
34 pipeline phase; and

35 nullifying a predicate instruction for a following execution
36 phase by operating said corresponding functional unit at a reduced
37 power state relative to normal instruction operation if said
38 corresponding predicate register has said second state during a
39 prior instruction decode pipeline phase and said corresponding
40 scoreboard bit has said second state during a prior instruction
41 decode pipeline phase.

1 5. (Original) The method of claim 4, further comprising the
2 step of:
3 resetting a scoreboard bit to a second digital state upon
4 nullification of said instruction designating said corresponding
5 data register as a destination operand data register.

6. (Canceled)

1 7. (Previously Presented) The method of claim 4 further
2 comprising the steps of:
3 statically scheduling instruction execution via a compiler;
4 and
5 scheduling via said compiler a last write to a data register
6 before an instruction decode pipeline phase of a predicate
7 instruction designating said data register as a predicate register.

1 8. (Previously Presented) The pipelined data processor of
2 claim 1, wherein:
3 each functional unit is operable at said reduced power state
4 by not fetching at least one instruction operand and not toggling a
5 corresponding register read port during said following execution
6 phase.

1 9. (Previously Presented) The pipelined data processor of
2 claim 1, wherein:

3 each functional unit is operable at said reduced power state
4 by not powering said functional unit during said following
5 execution phase.

1 10. (Previously Presented) The method of claim 4, wherein:
2 said step of operating said corresponding functional unit at a
3 reduced power state includes not fetching at least one instruction
4 operand and not toggling a correspond register read port during
5 said following execution phase.

1 11. (Previously Presented) The method of claim 4, wherein:
2 said step of operating said corresponding functional unit at a
3 reduced power state relative includes not powering said functional
4 unit during said following execution phase.

1 12. (New) The pipelined data processor of claim 1, wherein:
2 said scoreboard bit corresponding to each data register
3 capable of serving as a predicate register includes

4 an OR gate having a first input receiving a signal from a
5 corresponding functional unit indicating when a write
6 instruction to said corresponding predicate register commits,
7 a second input receiving a signal from said corresponding
8 functional unit indicating said write instruction to said
9 corresponding predicate register nullifies and an output, and

10 a flip-flop having a set input receiving an input from
11 said instruction decode unit indicating when said
12 corresponding predicate register is a destination for said
13 write instruction, a reset input connected to said output of
14 said OR gate and a Q output indicating a state of said
15 scoreboard bit.

1 13. (New) The pipelined data processor of claim 1, wherein:
2 each of said plurality of functional units includes

3 a compare to zero unit receiving data from said
4 instruction identified predicate register generating a signal
5 when data stored in said instruction identified predicate
6 register is zero,

7 a first AND gate having a first input receiving said
8 output of said compare to zero unit, a second input receiving
9 a signal indicating when said predicated instruction is in
10 said instruction decode pipeline phase, an inverting input
11 receiving said state of said corresponding scoreboard bit and
12 an output, and

13 a second AND gate having a first input receiving said
14 output of said compare to zero unit, a second input receiving
15 a signal indicating when said predicated instruction is in
16 said execution pipeline phase;

17 each of said plurality of functional units operable to

18 nullify said predicate instruction of a following
19 execution phase by operating at a reduced power state relative
20 to normal instruction operation upon generation of a signal at
21 said output of said first AND gate, and

22 not write said result of said predicate instruction of a
23 current execution phase upon generation of a signal at said
24 output of said second AND gate.